Econometrics and Statistics xxx (xxxx) xxx

Contents lists available at ScienceDirect

Econometrics and Statistics

journal homepage: www.elsevier.com/locate/ecosta

Bias correction for Vandermonde low-rank approximation

Antonio Fazzi[a,*], Alexander Kukush[b], Ivan Markovsky[a]

[a] Department ELEC, Vrije Universiteit Brussel (VUB), Pleinlaan 2, 1050 Brussels, Belgium
[b] Faculty of Mechanics and Mathematics, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

ARTICLE INFO

Article history:
Received 15 December 2020
Revised 3 September 2021
Accepted 3 September 2021
Available online xxx

Keywords:
Vandermonde matrix
Bias removal
Adjusted least squares
Structured matrix perturbation

ABSTRACT

The low-rank approximation problem, that is the problem of approximating a given matrix with a matrix of lower rank, appears in many scientific fields. In some applications the given matrix is structured and the approximation is required to have the same structure. Examples of linear structures are Hankel, Toeplitz, and Sylvester. Currently, there are only a few results for nonlinearly structured low-rank approximation problems. The problem of Vandermonde structured low-rank approximation is considered. The high condition number of the Vandermonde matrix, in combination with the noise in the data, makes the problem challenging. A numerical method based on a bias correction procedure is proposed and its properties are demonstrated by simulation. The performance of the method is illustrated on numerical results.

1. Introduction

The approximation of a matrix with another one which satisfies some constraints, e.g., rank conditions or structure preserving properties, is very popular because of applications in several scientific fields (Markovsky, 2019). In particular, the need to preserve the structure of the data matrix on the low-rank approximated solution leads to the formulation of nonconvex optimization problems, which do not admit analytic solutions as well as standard numerical solution methods. Several algorithms exist in the case of linearly structured matrices (matrices whose entries are linear functions of a parameter vector). Examples of linear structures are Hankel, Toeplitz or Sylvester and some of the corresponding structured low-rank approximation algorithms are presented in Fazzi et al. (2021); Markovsky (2019); Usevich and Markovsky (2014); Chu et al. (2003); Ishteva et al. (2014); Van Huffel et al. (1996) and in some references therein.

However if the entries of the involved matrices depend nonlinearly on the problem parameters, usually it is not possible to extend the same numerical approaches used in the linear case (few examples of existing generalizations from the linear case are Rosen et al. (1998); Lemmerling et al. (2002)) Therefore, the literature for nonlinearly structured low-rank approximation problems is quite poor nowadays.

The problem considered is approximation of a noisy Vandermonde matrix with another Vandermonde matrix of lower rank. The Vandermonde structure and the solution of Vandermonde systems of equation arise in several applications in exponential data modeling (Barkhuijsen et al., 1987; Rosen et al., 1998). Given a vector $c = (c_1, \ldots, c_n) \in \mathbb{C}^n$, the matrices

* Corresponding author.
E-mail addresses: Antonio.Fazzi@vub.be (A. Fazzi), alexander_kukush@univ.kiev.ua (A. Kukush), ivan.markovsky@vub.be (I. Markovsky).

https://doi.org/10.1016/j.ecosta.2021.09.001
2452-3062/

Please cite this article as: A. Fazzi, A. Kukush and I. Markovsky, Bias correction for Vandermonde low-rank approximation, Econometrics and Statistics, https://doi.org/10.1016/j.ecosta.2021.09.001

we consider are rectangular Vandermonde matrices

$$V = V(c) = \begin{pmatrix} 1 & \cdots & 1 \\ c_1 & \cdots & c_n \\ c_1^2 & \cdots & c_n^2 \\ \vdots & \cdots & \vdots \\ c_1^m & \cdots & c_n^m \end{pmatrix}, \tag{1}$$

where the (i, j)-th entry of the matrix is the $(i$-1$)$-th power of the j-th entry of the vector c (whose entries are also referred as problem parameters throughout the paper). Under the assumption that $c_i \neq c_j$ for all $i \neq j$, the matrix V has full rank, so that the problem of low-rank approximation is aiming at coalescing two of the parameters with as small as possible perturbation. The formulation of the problem is as follows:

Problem 1. Given a matrix of the form $V(c_0) + V(\tilde{c})$, where $V(c_0)$ is a rank-deficient Vandermonde matrix and $V(\tilde{c})$ is a Vandermonde Gaussian error matrix, we want to estimate the vector c_0.

The Vandermonde structure is nonlinear, so the considered problem is a nonlinearly structured low-rank approximation problem.

While structured low-rank approximation problems are popular and widely studied in the literature in the case of linear (or affine) structures (see e.g. Usevich and Markovsky (2014); Chu et al. (2003)), there are no general efficient methods for solving the nonlinearly structured low-rank approximation problems.

The proposed algorithm computes a solution for Problem 1 into two steps: first we compute an *adjusted* unbiased estimator $\hat{X}$ (Masiuk et al., 2017; Cheng, C.-L. and Schneeweiss, 1998) which *corrects* the classical (biased) solution of the optimization problem $\|VX\|_2^2 \to \min$ such that $\|X\|_2 = 1$, where $V \in \mathbb{C}^{(m+1)\times n}$ is the given Vandermonde matrix and $X \in \mathbb{C}^{n\times 1}$. The estimator $\hat{X}$ is an exact solution of a corrected optimization problem $X^* g(V) X \to \min$ such that $\|X\|_2 = 1$, where $g(V)$ is a correction of the matrix V^*V to the measurement errors contained in the error matrix $\tilde{V} = V(\tilde{c})$. Then, in order to preserve the Vandermonde structure of the data matrix on the computed solution, we need to estimate the parameters $\hat{c} \in \mathbb{C}^n$ starting from the estimated values $g(V), \hat{X}$ coming from the bias removal approach. The computed solution, however, is not expected to be the smallest parameters perturbation which gives an admissible solution (the algorithm is not optimization based), so the solution is only suboptimal in this sense.

Remark 1. The proposed methodology can be generalized also in the case when only a subset of the columns of V only needs to preserve the Vandermonde structure.

Applications

Problems of *fitting conic sections to data* (Markovsky et al., 2004; Shklyar et al., 2007; Markovsky and Usevich, 2014) have been posed and solved as a nonlinearly structured low-rank approximation problems with polynomial structure. The considered problem (Vandermonde structured low-rank approximation) is motivated from an application in computer algebra, that is finding an approximate common factor for a set of given polynomials.

Let $r_1 \in \mathbb{C}^n$ be the set of roots of a polynomial $p_1(z)$ and $r_2 \in \mathbb{C}^m$ be the set of roots of $p_2(z)$. Assuming that all the roots have multiplicity one, these polynomials have no common factors if and only if the entries of r_1 and r_2 are pairwise different. Conversely, the presence of one (or more) polynomial common root means that the polynomials p_1, p_2 have a non-trivial common factor. This is because, under the previous assumption, we can factor the polynomials as products of degree-one factors. Therefore, given two vectors r_1 and r_2 whose entries are all different, the computation of a closest pair of vectors $\hat{r}_1, \hat{r}_2$ having (at least) one common entry gives a pair of perturbed polynomials having a common factor.

The parameter c generating a Vandermonde matrix can be the set of noisy roots of two polynomials. Since the rank deficiency of a Vandermonde matrix means that $\hat{c}_i = \hat{c}_j$ for some i, j, Problem 1 can be seen as a computation of distance to common divisibility for the polynomials p_1, p_2 (we only need to pay attention that $\hat{c}_i$ and $\hat{c}_j$ are roots of different polynomials). The problem is known in the literature also as approximate greatest common divisor or approximate common factor computation. The importance of the problem is due to its applications in the fields of system and control theory and identification (Markovsky et al., 2018; De Iuliis et al., 2017).

However, the usual definition of distance to common divisibility considers a measure based on the polynomials coefficients. This allows to restate the problem as a Sylvester matrix low-rank approximation problem, see Guglielmi and Markovsky (2017); Markovsky and Van Huffel (2006); Pan (2001); Qiu et al. (1997); Terui (2013); Usevich and Markovsky (2014); Zeng (2011); Fazzi et al. (2019).

The formulation of a Vandermonde structured low-rank approximation problem, on the other hand, leads to a different definition of distance based on the roots of the polynomials. Note that the polynomials roots are uniquely defined while the coefficients are not (indeed, if we multiply the polynomials by a nonzero constant, we change the coefficients of the polynomials but not their roots).

Another possible application to exponential data modeling is presented in Rosen et al. (1998, Section 4). The only difference with Problem 1 is that only A is a Vandermonde matrix in the solution of a system of the form $AX = B$.

Main contribution

In the following we propose a suboptimal approach for the solution of Problem 1. In the first part we construct a *corrected* estimator for the solution of the system of equations $VX = 0$, where the matrix V is a noisy Vandermonde matrix. The proposed estimator builds an exact solution $\hat{X}$ of the system of equations $\hat{V}X = 0$, where $\hat{V}$ is a correction of the matrix V necessary to remove the common least squares bias. This is closely related to the total least squares problem since we look for an exact solution of a corrected system of equation. The difference with respect to the classical total least squares formulation is that we do not compute the smallest norm perturbation, but we approach the problem using some deconvolution equations (see Section 2 for details).

Then, we observe that we did not impose any structure constraint on the previous estimator of $\hat{V}$ but we need the Vandermonde structure in the final solution. Hence, a further step is needed in order to estimate the perturbed parameter vector $\hat{c}$ (the parameters generating the Vandermonde matrix $V(\hat{c})$) starting from the solution of the corrected system of equations.

Notation

We denote with D^* the Hermitian adjoint of the matrix D and with D^+ the pseudo-inverse of D. $\bar{c}$ is the complex conjugate of c. $\mathbf{E}(\cdot)$ denotes the expected value of the corresponding random variable, random vector or random matrix. Two exclamation marks denote the double factorial.

In what follows, the "exact data" $V(c_0)$ is a Vandermonde matrix generated by a vector $c_0 \in \mathbb{C}^n$ whose entries are on the unit circle and such that $c_0(l) = c_0(m)$ for some indexes l, m. $V_0 = V(c_0)$ is the corresponding (rank-deficient) Vandermonde matrix. The "noisy data" $V(\tilde{c})$ is another Vandermonde matrix generated by a vector $\tilde{c} \in \mathbb{C}^n$ of the form

$$\tilde{c}_j = \tilde{u}_j + i\tilde{v}_j, \quad \text{for all } j = 1, \dots, n, \tag{2}$$

where $\tilde{u}_j, \tilde{v}_j, j = 1, \dots, n$ are independent identically distributed (i.i.d.), zero mean, Gaussian with standard deviation σ.

Remark 2. The normalization assumption on the "exact data" (that is the entries of c_0 to be on the unit circle) may seem restrictive, but it is necessary to avoid the vanishing/blow up of the entries of the Vandermonde matrix whenever a large number of rows (high exponents of a power) is considered. Even if we impose the constraint $|c_0(j)| = 1$ for all j, we assume this information is not known in advance, so the estimation method also computes the magnitudes of the sought parameters.

2. Construction of adjusted least squares estimator

In this section we deal with a numerical solution of the system of equations $VX = 0$. A possible approach can be to partition the columns of V into two subsets A and B and deal with a system of equations of the form $AX = B$. A classical solution technique for this uses the Total Least Squares solution, which yields (under standard assumptions) a maximum likelihood estimator (Markovsky and Van Huffel, 2007). The proposed approach avoids partitioning the matrix V and it is based on the solutions of deconvolution equations (Masiuk et al., 2017; Cheng, C.-L. and Schneeweiss, 1998) which guarantee that the corresponding estimators are unbiased. We analyze both the cases in which the value of σ (the standard deviation of the normal perturbation on the *exact* data, see (2)) is known and unknown. In particular, we describe in details how to build the proposed estimator in the case when σ is known, and we explain how to estimate σ in the case when it is unknown.

2.1. Known σ^2

We assume here that σ is known, and we want to compute an unbiased estimator $\hat{X}$ which corrects the classical biased solution of the optimization problem $\|VX\|_2^2 \to \min$ such that $\|X\|_2 = 1$, where $V \in \mathbb{C}^{(m+1)\times n}$ is the given Vandermonde matrix and $X \in \mathbb{C}^{n\times 1}$. The idea comes from the Orthogonal Regression Estimator in the implicit linear model in $\mathbb{R}^n$ (see Masiuk et al. (2017, Section 2.5.3)). If we know that the noiseless matrix $V_0 = V(c_0)$ is rank-deficient, then there exists a vector $X \in \mathbb{C}^{n\times 1}$, $\|X\|_2 = 1$, with $V_0X = 0$. Thus, $X = (e_i - e_j)/\sqrt{2}$, where $c_0(i) = c_0(j), i \neq j$, and $e_k, 1 \le k \le n$, is the k-th vector in the canonical base of $\mathbb{R}^n$. Therefore the vector X minimizes the following cost function

$$Q_0(V_0; X) = X^*V_0^*V_0X, \qquad X \in S_{n-1} = \{Y \in \mathbb{C}^{n\times 1} : \|Y\|_2 = 1\}. \tag{3}$$

The idea of bias removal leads to the solution of the following problem: given the matrix V_0, compute a corrected cost function which solves a deconvolution equation of the form

$$\mathbf{E}(Q_c(V; X)|V_0) = Q_0(V_0; X). \tag{4}$$

Looking at the function in (3), the main idea is to correct the cost function such that (4) holds true. This is based on the method of corrected cost functions (Kukush and Zwanzig, 2001). The solution is given by the following deconvolution equation:

$$\mathbf{E}(g(V)|V_0) = V_0^*V_0. \tag{5}$$

The next theorem suggests how to get the sought solution.

Theorem 1. *Remembering that* $\tilde{V} = V(\tilde{c})$, *define the covariance matrix*

$$K_{\tilde{V}} = \mathbf{E}((\tilde{V} - \mathbf{E}(\tilde{V}))^*(\tilde{V} - \mathbf{E}(\tilde{V}))) = \mathbf{E}(\tilde{V}^*\tilde{V}) - \mathbf{E}(\tilde{V}^*)\mathbf{E}(\tilde{V}).$$

The solution of the equations in (5) is given by the function

$$g(V) = (V - \mathbf{E}(\tilde{V}))^*(V - \mathbf{E}(\tilde{V})) - K_{\tilde{V}}. \tag{6}$$

Proof. The function $g(V)$ can be expanded as

$$g(V) = V^*V - \mathbf{E}(\tilde{V}^*)V - V^*\mathbf{E}(\tilde{V}) + \mathbf{E}(\tilde{V}^*)\mathbf{E}(\tilde{V}) - K_{\tilde{V}}. \tag{7}$$

By computing the expected value of each term in (7) we get

$$\begin{aligned} &\mathbf{E}(V^*V|V_0) = V_0^*V_0 + \mathbf{E}(\tilde{V}^*)V_0 + V_0^*\mathbf{E}(\tilde{V}) + \mathbf{E}(\tilde{V}^*\tilde{V}) \text{ and} \\ &\mathbf{E}(V|V_0) = V_0 + \mathbf{E}(\tilde{V}). \end{aligned} \tag{8}$$

The claim follows by replacing (8) in (7) and summing up the correct terms. □

Then, the corrected cost function has the form

$$Q_c(V; X) = X^*g(V)X, \quad X \in S_{n-1}. \tag{9}$$

Observe that the function in (9) satisfies (4) for all vectors $X \in S_{n-1}$. The sought estimator $\hat{X}$ can be computed by minimizing the cost function Q_c over all the vectors in S_{n-1}. Since $g(V)$ is symmetric and positive semi-definite by construction, the estimator $\hat{X}$ is a normalized eigenvector of $g(V)$ corresponding to the smallest eigenvalue, that is

$$\min_{X \in S_{n-1}} Q_c(X) = \lambda_{\min}(g(V)). \tag{10}$$

$\hat{X}$ is the eigenvector associated with $\lambda_{\min}(g(V))$.

Once we have found the expression of the adjusted unbiased estimator, we switch to the computational issue: how to build the matrix $g(V)$.

Computation of covariance matrices

We show in the following how to build the matrices $\mathbf{E}(V), K_{\tilde{V}}$ necessary to build the function g in (6) and consequently the estimator $\hat{X}$. Given σ, the entries of the matrix $K_{\tilde{V}}$ (defined in Theorem 1) are polynomials in the parameters $\tilde{c}_j$ because of the Vandermonde structure (1) of the matrix V. The expected value of a polynomial is split into the expected values of monomials; in particular a monomial of the form $\tilde{c}_k^p c_l^q$ has expected value

$$\begin{aligned} \mathbf{E}((u_k - iv_k)^p(u_l + iv_l)^q) &= \sum_{j=0}^{p}\sum_{t=0}^{q} \tilde{c}_k^j c_l^t (-i)^{p-j} i^{q-t} \mathbf{E}(u_k^j u_l^t)\mathbf{E}(v_k^{p-j} v_l^{q-t}) \\ &= \sigma^{p+q} \sum_{j=0}^{p}\sum_{t=0}^{q} \tilde{c}_k^j c_l^t (-i)^{p-j} i^{q-t} \mathbf{E}(\gamma_k^j \gamma_l^t)\mathbf{E}(\gamma_k^{p-j} \gamma_l^{q-t}), \end{aligned} \tag{11}$$

where all the γ_i are i.i.d. random variables coming from a normal distribution with zero mean and unit variance. Their expected values are

$$\begin{aligned} &\mu_0 = \mathbf{E}(\gamma_1^0) = 1 \quad \text{and} \\ &\mu_n = \mathbf{E}(\gamma_1^n) = \begin{cases} 0 & \text{if } n \text{ odd} \\ (2k-1)!! & \text{if } n \text{ even}, \quad n = 2k \end{cases}. \end{aligned} \tag{12}$$

Replacing (12) in (11) we get the following expressions:

$$\mathbf{E}(\tilde{c}_k^p c_l^q) = \begin{cases} \sigma^{p+q} \sum_{j=0}^{p}\sum_{t=0}^{q} \tilde{c}_k^j c_l^t (-i)^{p-j} i^{q-t} \mu_{j+t}\mu_{p+q-j-t} & \text{if } k = l \\ \sigma^{p+q} \sum_{j=0}^{p}\sum_{t=0}^{q} \tilde{c}_k^j c_l^t (-i)^{p-j} i^{q-t} \mu_j\mu_t\mu_{p-j}\mu_{q-t} & \text{if } k \neq l \end{cases}. \tag{13}$$

Hence, the covariance matrix $K_{\tilde{V}}$ is obtained as the sum of terms of the form (13). As we see from (13), each term depends on σ, the powers p, q and the indices k, l. By setting $p = 0$ in (13) we automatically find the expression for $\mathbf{E}(c_l^q)$, which are the expected values of the entries of a Vandermonde matrix: these values are the entries of the matrix $\mathbf{E}(V)$.

Remark 3. When q is odd, we have $\mathbf{E}(c_l^q) = 0$; this is because of the symmetry of the centered Gaussian distribution.

Note that the previous formula refers to Gaussian noise; each kind of error distribution has its own formula depending on the expression of the moments.

2.2. Unknown σ^2

We saw in the previous section how to build the estimator for given σ; therefore, in the case σ is unknown, we focus only on the estimation of σ. Once σ is estimated, we plug-in the estimate to the solution (10).

In this case, both the function g and the (corrected) penalty function Q_c depend also on σ:

$$\begin{aligned} &g(V) = g(V, \sigma^2) \quad \text{and} \\ &Q_c(X) = Q_c(X, \sigma^2). \end{aligned} \tag{14}$$

As in the previous case, the corrected penalty function Q_c is unbiased in the sense that

$$\mathbf{E}(Q_c(V; X)|V_0) = Q(V_0; X). \tag{15}$$

Similar computation to the previous case lead to the solution of the problem

$$\min_{X \in S_{n-1}} Q_c(X, \sigma^2) = \lambda_{\min}(g(V, \sigma^2)),$$

where the eigenvector $\hat{X}$ associated with $\lambda_{\min}(g(V, \sigma^2))$ is the sought estimator. Observe that σ^2 is a variable of the problem.

But we saw in Section 2 that the estimator of X actually depends on σ too, so we can write

$$\inf_{X \in S_{n-1}, \sigma^2} Q_c(X, \sigma^2) = \inf_{\sigma^2} \inf_{X \in S_{n-1}} Q_c(X, \sigma^2) = \inf_{\sigma^2} Q_c(\hat{X}_{\sigma^2}, \sigma^2), \tag{16}$$

where $\hat{X}_{\sigma^2} = \hat{X}|\sigma^2$ denotes the value of the estimator $\hat{X}$ for a given value of σ^2; moreover it is reasonable to consider only the values of σ^2 for which the matrix $g(V)$ is positive semidefinite, as it happens for the matrix $V_0^* V_0$. Finally, the estimation of σ ends up with the solution of the (univariate) optimization problem

$$\hat{\sigma}^2 = \inf_{\substack{\sigma^2 \\ g(V) \text{ positive semidefinite}}} Q_c(\hat{X}_{\sigma^2}, \sigma^2). \tag{17}$$

Remark 4. The numerical solution of (17) is computed by sampling different values of σ and choosing the one which minimizes the function Q_c (9) under the constraint on $g(V)$. This assumes implicitly that suitable lower and upper bounds for the true value of σ are known in advance.

2.3. About the estimator

We describe here the behavior of the proposed estimator $\hat{X}$. First of all, we have to mention that both the error of the estimator and the estimate of $\hat{\sigma}$ depend on the dimension m of the matrix $V(c)$ (1) (for n fixed), as well as on the noise distribution. Indeed, we would expect that bigger values of m (more information on the data) correspond to better approximations in terms of the computed solution (if X is the solution of $V_0 X = 0$, the error on the estimator $\hat{X}$ is measured via the relative distance $\|\hat{X} - X\|_2 / \|X\|_2$). But we see in Fig. 1 what happens if we run a numerical experiment with $n = 4$ parameters and increasing values of m ($\sigma = 10^{-1}$ is known).

The error decreases only up to a certain value of m while we observe something unexpected in the final part. What we see in Fig. 1 is a consequence of the fact that, in the Vandermonde structure, high values of m cause the function in (12) to blow up in machine arithmetic), while we would like the value of $\hat{X}$ to approach the exact solution of the system $V_0 X = 0$ as $m \to \infty$. The noisy entries of the Vandermonde matrix can blow up or vanish for high powers and for some perturbation values on the noiseless data. The observed result is actually due to the choice of the Gaussian noise on the data. It happens that the Gaussian noise makes the Vandermonde signal vanishing under the perturbation, and this is because the moments of order $2k$ of Gaussian random variables grow with k faster than the exponent of k; therefore, the solution associated to the Vandermonde system of equations can vanish under the considered perturbation.

Remark 5. If the Gaussian distribution of the noise is replaced by a symmetrical distribution with compact support, the results for the error of the estimator $\hat{X}$ would be different with respect to the ones shown in Fig. 1. This is because the error moments of order $2k$ would grow not faster than the exponent of k in this case. However, we believe that the noise with compact support (e.g., uniformly distributed noise) is not realistic in an application setting, hence the choice of the Gaussian noise.

From the previous experiment we observe that, up to a certain point, the increasing of the value of m actually leads to an improvement in the quality of the estimator. However, the numerical values suggest that such improvement is not so relevant, so a wide range of values of m should lead to numerical results which are very close.

3. Estimation of the parameters

The algorithm described in Section 2 only computes $\hat{X}$ (and it estimates σ if this is unknown); however the corrected matrix $\hat{V}$ (which is not computed explicitly but it is given through the function $g(V)$) does not preserve the Vandermonde

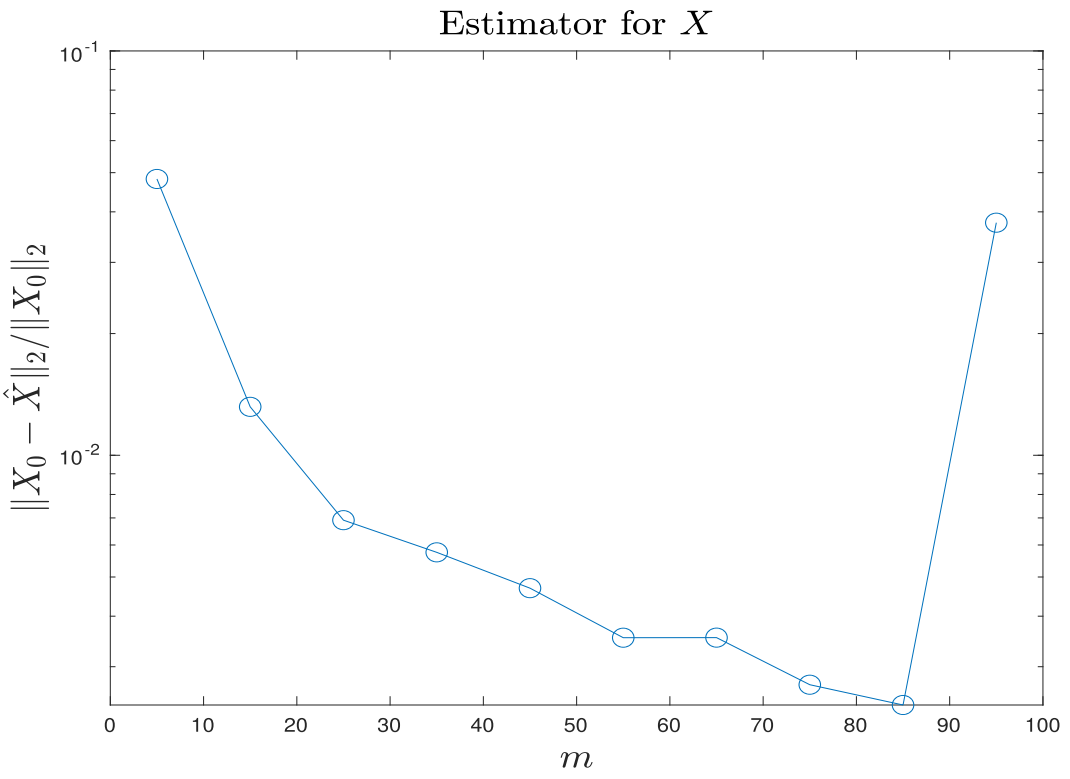

Fig. 1. Relative distance between the estimator $\hat{X}$ and the true value X_0 for increasing values of m: error averaged over 50 different perturbation on the same data.

structure (1), hence we need to refine the computed solution in order to get a perturbed parameter vector $\hat{c}$ generating a rank-deficient Vandermonde matrix. The goal of this section is to estimate the sought Vandermonde matrix $V(\hat{c}), \hat{c} \in \mathbb{C}^n$ starting from the computed solution $g(V), \hat{X}$. We list in Algorithm 1 the computational procedure.

Before running some experiments, we make some comments on the different steps in Algorithm 1. (19) comes from the structure of the matrix $\hat{D}$. In particular, the i-th diagonal entry of the square matrix $\hat{D}$ is a univariate polynomial in the variable c_i. If $\hat{d}_{jj} > 1$, (19) has a unique real solution (we are looking for a modulus, that is a positive real number); if $\hat{d}_{jj} < 1$ we set the corresponding $\hat{r}_j$ equal to zero. Once we estimate the magnitudes (and possibly we sort them together with the columns of $\hat{D}$), we need to guess the phase of the sought parameters by using the matrix $\hat{D}$. (20) may have no exact solution, so we consider as solution the value $\hat{\phi}$ corresponding to the smaller error. The non-uniqueness of this (approximate) solution of (20) is mitigated by the last steps of the algorithm, where we try to *refine* the computed solution by getting closer to the starting parameters. The solution of (21) seeks to *correct* the estimated phases by using $\hat{X}$ and multiplying all the parameters by a common factors. The last step guarantees the rank constraint on the Vandermonde matrix.

Remark 6. We note that it is not easy to recover exactly some complex numbers α, β from a polynomial value $p(\alpha, \beta)$. While the estimation of the modulus is an easier problem, the estimation of the phase is more complicated also in the noise-free case (we actually did not run the algorithm at all if the starting matrix is rank-deficient). This is due to the sampling of the interval $(0, 2\pi)$, or in general to the several numerical approximations due to the machine arithmetic.

4. Numerical experiments

In this section, we report results of numerical experiments that validate the algorithm and illustrate the performance of the proposed estimation method. We do experiments on both the estimation of σ and the parameters estimation. The setup for the examples is as follows:

1. we generate $n-1$ (random) complex roots $c_1, \ldots, c_{n-1}$ on the unit circle and then we repeat one of them (we do not known in advance which is the repeated root); the vector $(c_1, c_1, \ldots, c_{n-1})$ generates a rank-deficient Vandermonde matrix;
2. we perturb the starting Vandermonde matrix by adding a second Vandermonde matrix generated by a random complex vector whose entries are i.i.d. white Gaussian coming from a distribution of the form $N(0, \sigma^2)$;

Algorithm 1: Algorithm for Vandermonde low-rank approximation using an unbiased estimator

Data: V (and possibly σ)
Result: $\hat{c}$ (estimated parameters)
begin

1 Compute an unbiased estimator $\hat{X}$ and a corrected matrix $g(V)$ using the method in Section 2

2 Set $\hat{D} = g(V, \hat{\sigma}^2)$: by construction, $\hat{D}$ satisfies the deconvolution equation

$$\mathbf{E}(\hat{D}|V_0) = V_0^* V_0. \tag{18}$$

3 Estimate the modulus $\hat{r}_j = |\hat{c}_j|$ for all j. If we denote as $\hat{d}_{ij}$ the entries of $\hat{D}$, using (18), we need to solve the equation

$$\hat{d}_{jj} = \sum_{k=1}^{m+1} \hat{r}_j^{2(k-1)} \quad \forall j. \tag{19}$$

4 Sort $\hat{r}$ by increasing order. Set $\hat{r}_{\bar{j}} = \max_j \hat{r}_j$. Assuming $r_{\bar{j}} > 0$,
$\hat{c}_{\bar{j}} = r_{\bar{j}}$
$\hat{c}_j = \exp(i\hat{\phi})\hat{r}_j,\ j \neq \bar{j}$
where $\hat{r}_j$ comes from the solution computed at step 2, while $0 \leq \hat{\phi} < 2\pi$ is a solution of the equation

$$\hat{d}_{\bar{j}j} = \sum_{k=1}^{m+1} (\hat{r}_{\bar{j}}\hat{r}_j)^{k-1} \exp(i\hat{\phi}_j(k-1)) \quad \forall j \neq \bar{j}. \tag{20}$$

Observe that the solution of this last equation is not unique

5 Set $\hat{c}_j = \hat{r}_j \exp(i\hat{\phi}_j) \exp(i\hat{\theta}) \quad \forall j$
where $\hat{\theta} \in (0, 2\pi)$ solves the optimization problem

$$\min_{\hat{\theta}} \|V - \mathbf{E}(V) - V(\hat{c}(\hat{\theta}))\|_F \tag{21}$$

where $V(\hat{c})$ is the Vandermonde matrix generated by the parameter vector $\hat{c}$

6 Minimize the distance $\|c - \hat{c}\|_2$ by possibly changing the sign and/or swapping the real and imaginary parts of each $\hat{c}_j$

7 Find $p \neq q$ such that

$$|\hat{c}_p - \hat{c}_q| = \min_{1 \leq j,k \leq n,\ j \neq k} |\hat{c}_j - \hat{c}_k|.$$

Set $\hat{c}_p = \hat{c}_q = (\hat{c}_p + \hat{c}_q)/2$.

3. we show the estimation results averaged over different perturbations (of increasing noise levels) on the starting parameters.

The standard deviation of the noise distribution varies in the interval $(0, 0.3)$; the considered estimation error is the relative distance between the computed parameters $\hat{c}$ and the starting parameters c: $\|c - \hat{c}\|_2 / \|c\|_2$. We run each time several experiments with different perturbations and the same noise levels on the parameters c_0, and we consider the average error (this is to reduce the effect of misleading results).

4.1. Estimation of σ^2

We check first how the estimate of σ^2 performs in the case σ is unknown. In this experiment we focus on the estimation of σ only, by solving the univariate optimization problem (16), while we deal with the parameters estimation later. This problem is associated with the estimator built in Section 2. In the following experiment we compare the true value of σ (the one we used to generate the data) with the estimated one. We consider here a matrix with $m = 25$ rows since the experiment would be time consuming otherwise; however we observed in Fig. 1 that the estimation error can be decreased by increasing the value of m even if the improvement is not so relevant.

We did not consider bigger values of m since the experiment is time consuming: each time we run an experiment with known σ, for all $\sigma \in (0, 0.5)$, by using the available data and then we solve the optimization problem (17).

We observe that, for small values of σ (approximately the first half of the plot), the estimated values are very close to the exact ones. The estimates become worse for higher values of σ, but the difference from the exact values is not so high.

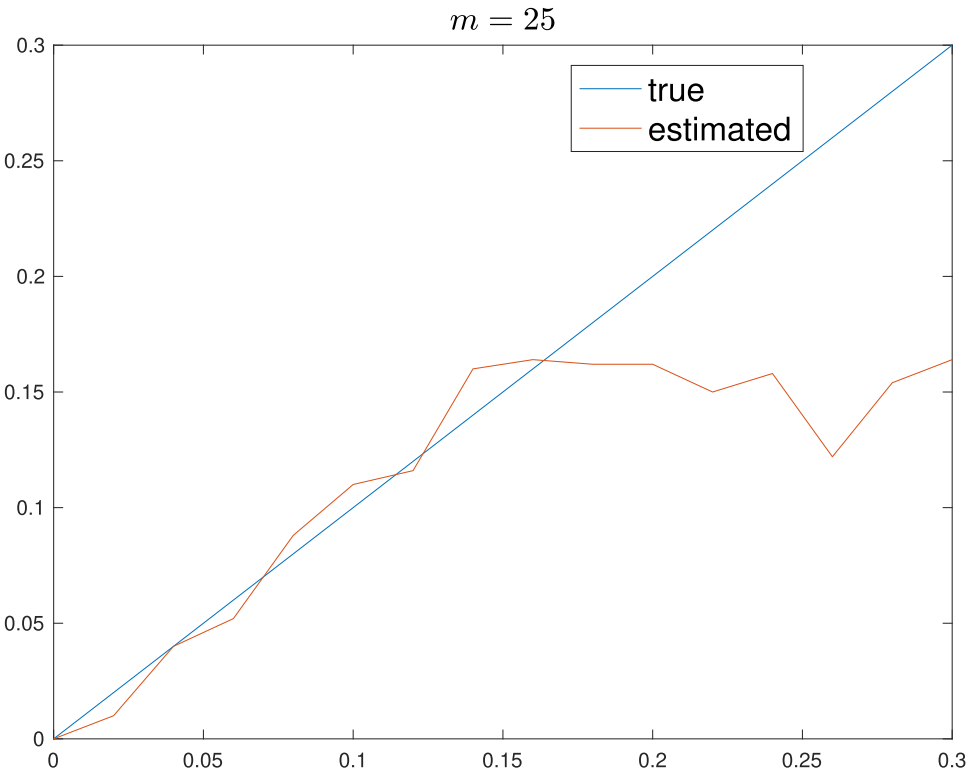

Fig. 2. Estimation of σ: result averaged over 10 different perturbation on the data associated with normal distributions with the same standard deviation.

4.2. Parameters estimation

Once we know the value σ^2 (it can be known or estimated through the solution of (16)) we can run some experiments using Algorithm 1 in order to test the parameters estimation. In the experiments we fix the number of parameters $n = 4$, and we run the estimation algorithm for different perturbations corresponding to increasing values of σ (in the interval $(0, 0.3)$) in order to observe how the output changes depending on the noise effect on the data.

We plot in the following the relative error on the parameters ($\|\hat{c} - c\|_2/\|c\|_2$) and the relative distance with respect to the noiseless data (this information is actually unknown to the problem, but we add it for information purposes). The computed Vandermonde matrices are always rank-deficient by construction. The following figures show the results for 4 parameters with $m = 5$ (Fig. 3).

We see how the error slightly increases with σ. The value of m is small in order to avoid slow computations with big matrices (observe we ran 500 experiments for each σ). By comparing the two plots, we observe that they behave in a similar way. The high values for the relative error, even for small values of σ, are probably due to the high condition numbers of Vandermonde matrices, which combined with the nonlinear nature of the estimation procedure, make the problem challenging.

We wonder what happens if we increase the value of m, that is more information available for the same data. We saw in Fig. 1 that the error on the estimator $\hat{X}$ can decrease for increasing values of m (but only in a certain interval), so it makes sense to investigate what happens for the estimated parameters. We show in Fig. 4 a simulation for two different values of m (the two Vandermonde matrices are generated from the same parameter vector but their dimensions are different). We see that, except for high values of σ, the two plots are very close but bigger values of m correspond to bigger errors on the estimated parameters with respect to both the noisy and the noiseless data. Such a distance increases for higher values of σ. So we observe that increasing the value of m improves the value of the estimator $\hat{X}$ but not the accuracy on the estimated parameters. This is because the high powers in the Vandermonde structure amplify the errors on the data even if more equations lead to more information for the system solution. A trade-off between the two errors would probably be the right balance.

4.3. Approximate polynomial common factor computation

Finally we show a numerical example of distance to common divisibility for two polynomials, as described in the subsection *Applications* in Section 1. In particular, consider the following two complex polynomials of degree 2 (all the numerical

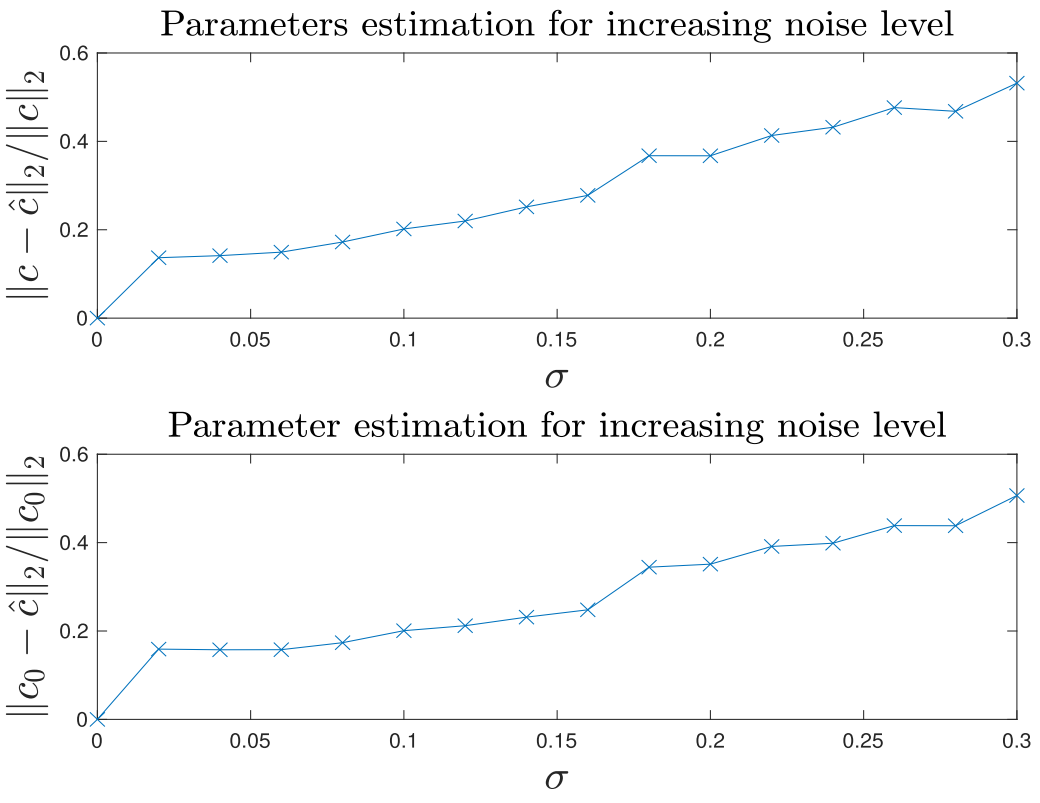

Fig. 3. Vandermonde low-rank approximation for 4 complex parameters. Relative error on the estimated parameters (top) and relative distance with respect to the noiseless data (bottom) averaged over 500 different perturbations.

values in this section are approximated to the fourth decimal digit):

$$\begin{aligned} p_1(z) &= z^2 + (0.8129 + 0.8498i)z + (-0.444 + 0.9990i) \quad \text{and} \\ p_2(z) &= z^2 + (0.1320 + 0.3778i)z + (-0.7825 + 0.6227i), \end{aligned}$$

whose roots are

$$\begin{aligned} r_1 &= (0.1781 - 0.9840i, -0.9910 + 0.1342i) \quad \text{and} \\ r_2 &= (0.8590 - 0.5120i, -0.9910 + 0.1342i). \end{aligned}$$

The starting (noiseless) parameter vector c_0 is the set of all roots $c_0 = (r_1, r_2)$. The data for the problem are then generated by perturbing the vector c_0 as in (2) using a normal distribution with $\sigma = 0.1$. The vector c whose entries are all different is

$$c = (0.2053 - 0.9661i, -1.0332 + 0.1033i, 0.8105 - 0.6361i, -1.1416 + 0.1573i).$$

We immediately observe that the polynomials $(z - c(1))(z - c(2))$ and $(z - c(3))(z - c(4))$ have no common roots. We apply then Algorithm 1 to recover two polynomials having a non-trivial common factor (we consider a Vandermonde matrix with 5 rows). The computed parameters $\hat{c}$ are

$$\hat{c} = (0.1903 - 0.9789i, -1.0018 + 0.1230i, 0.9943 - 0.1045i, -1.0018 + 0.1230i).$$

The second and the last entry of the vector $\hat{c}$ are the same, hence a common root (that is a common factor) between the polynomials has been found. However this is not exactly the same as the starting one (the common entry between r_1 and r_2) because the non-convexity and the nonlinearity of the problem make it difficult to compute an accurate solution of the associated structured low-rank approximation problem.

Remark 7. The estimation of the parameter in Algorithm 1 works in a similar way even when the noiseless data are not on the unit circle, as described in the paragraph *Notation* in Section 1. This may be useful in an application setting because there is no restriction on the problem data. However, we need to pay attention to the value of m in the Vandermonde matrix since the moduli of the data may cause the vanishing/blow up of the entries of the matrix in machine arithmetic. As a consequence the numerical results can be misleading because the error of the estimator is not guaranteed to be monotonically decreasing for increasing values of m.

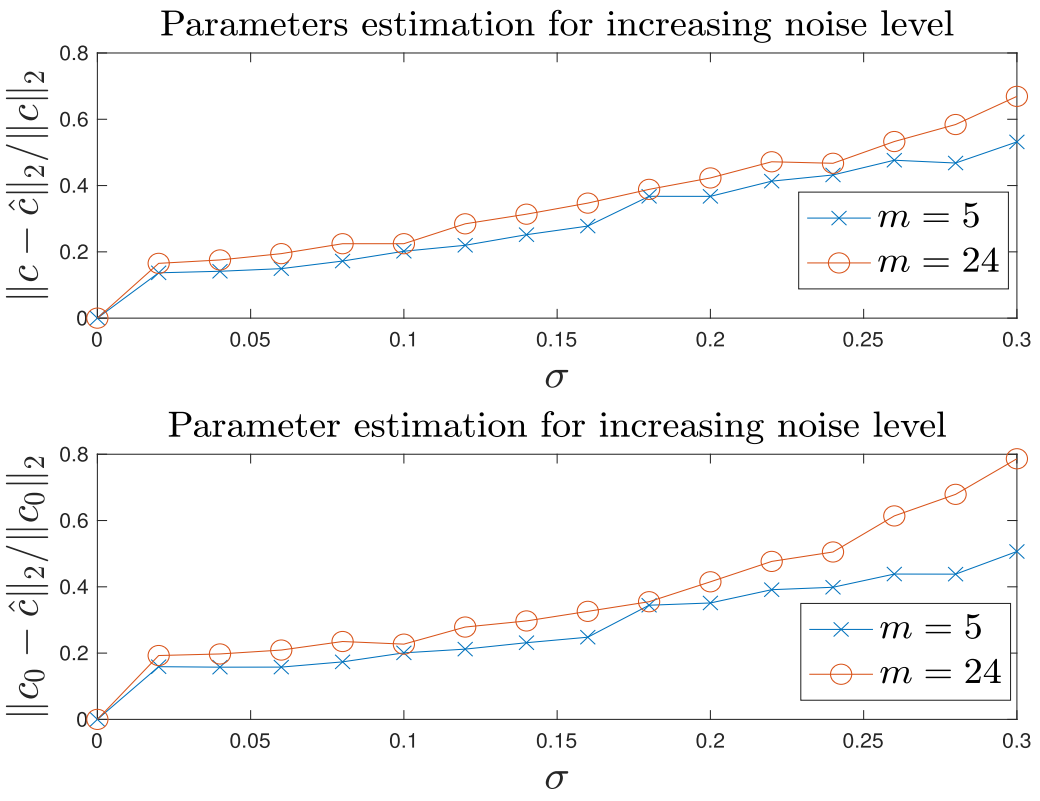

Fig. 4. Vandermonde low-rank approximation for 4 complex parameters and two different values of m. Relative error on the estimated parameters (top) and relative distance with respect to the noiseless data (bottom) averaged over 500 different perturbation.

5. Conclusion

The problem data is a rank-deficient Vandermonde matrix and, after the addition of some noise, a method for its recovery is proposed. This is based on the unbiased solution of a system of equations, followed by the estimation of the sought structured matrix. However, the problem is not so easy to deal with: if, on one hand, one can investigate different computational techniques for both the solution of the system of equations and the parameters estimation, on the other hand, the high condition numbers of Vandermonde matrices make hard the exact recovery of the parameters, even for small errors on the data.

The problem and its solution method were motivated by an application in computer algebra, that is approximate polynomial common factors computation, but the same algorithm can be used in all the applications which require the solution of (noisy) Vandermonde systems of equations. The only warning is related to the dimension of the Vandermonde matrix and the error of the estimator, especially when the magnitude of the data is arbitrary. The successive application of the method can be used if a rank reduction greater than one is needed. If the noiseless parameter vector c_0 has p repeated entries, in alternative, we can adapt both the steps of the algorithm by suitably changing the involved indices in order to compute a rank reduction by p.

Declaration of Competing Interest

The authors declare that there is no conflict of interest.

Acknowledgements

The research leading to these results has received funding from the Fond for Scientific Research Vlaanderen (FWO) projects G028015N and G090117N and the FNRS-FWO under Excellence of Science (EOS) Project no 30468160 “Structured low-rank matrix / tensor approximation: numerical optimization-based algorithms and applications” and also by the National Research Fund of Ukraine grant 2020.02/0026.

References

Barkhuijsen, H., De Beer, R., Van Ormondt, D., 1987. Improved algorithm for noniterative time-domain model fitting to exponentially damped magnetic resonance signals. J. Magentic Resonance 73, 553–557. doi:10.1016/0022-2364(87)90023-0.
Cheng, C.-L, Schneeweiss, H., 1998. Polynomial regression with errors in the variables. J. Royal Statist. Society Series B (Statist. Mathodol.) (60) 189–199. doi:10.1111/1467-9868.00118.
Chu, M.T., Funderlic, R.E., Plemmons, L.J., 2003. Structured low rank approximation. Linear Algebra Its Appl. 366, 157–172. doi:10.1016/s0024-3795(02)00505-0.
De Iuliis, V., Germani, A., Manes, C., 2017. Identification of forward and feedback transfer functions in closed-loop systems with feedback delay. IFAC-PapersOnLine 50 (1), 12847–12852. doi:10.1016/j.ifacol.2017.08.1935.
Fazzi, A., Guglielmi, N., Markovsky, I., 2019. An ODE based method for computing the Approximate Greatest Common Divisor of polynomials. Numer. Algorithms 81, 719–740. doi:10.1007/s11075-018-0569-0.
Fazzi, A., Guglielmi, N., Markovsky, I., 2021. A gradient system approach for Hankel structured low-rank approximation. Linear Algebra its Appl. 623, 236–257. doi:10.1016/j.laa.2020.11.016.
Guglielmi, N., Markovsky, I., 2017. An ODE based method for computing the distance of coprime polynomials to common divisibility. SIAM J. Numer. Anal. 55, 1456–1482. doi:10.1137/15M1018265.
Ishteva, M., Usevich, K., Markovsky, I., 2014. Factorization Approach to Structured Low-Rank Approximation with Applications. SIAM J. Matrix Anal. Appl. 35, 1180–1204. doi:10.1137/130931655.
Kukush, A., Zwanzig, S., 2001. On an adaptive estimator of the least contrast in a model with nonlinear functional relations. Ukr. Math. J. 53, 1445–1452. doi:10.1023/a:1014358423205.
Lemmerling, P., Van Huffel, S., De Moor, B., 2002. The structured total least-squares approach for non-linearly structured matrices. Numer. Linear Algebra Appl. 9, 321–332. doi:10.1002/nla.276.
Markovsky, I., 2019. Low-Rank Approximation, 2 Springer International Publishing.
Markovsky, I., Fazzi, A., Guglielmi, N., 2018. Applications of polynomial common factor computation in signal processing. In: et al., Y.D. (Ed.), Latent Variable Analysis and Signal Separation. Springer. Vol. 10891 of Lecture Notes in Computer Science, pp. 99–106., DOI: 10.1007/978-3-319-93764-9_10
Markovsky, I., Kukush, A., Van Huffel, S., 2004. Consistent least squares fitting of ellipsoids. Numerische Mathematik 98 (1), 177–194. doi:10.1007/s00211-004-0526-9.
Markovsky, I., Usevich, K., 2014. Nonlinearly structured low-rank approximation. In: Fu, Y.R. (Ed.), Low-Rank and Sparse Modeling for Visual Analysis, pp. 1–22. Springer. DOI: 10.1007/978-3-319-12000-3_1
Markovsky, I., Van Huffel, S., 2006. An algorithm for approximate common divisor computation. In: Proceedings of MTNS'06. Kyoto, Japan, pp. 274–279.
Markovsky, I., Van Huffel, S., 2007. Overview of total least squares methods. Signal Process 87, 2283–2302. doi:10.1016/j.sigpro.2007.04.004.
Masiuk, S., Kukush, A.G., Shklyar, S.V., Chepurny, M.I., Likhtarov, I.A., 2017. Radiation Risk Estimation: Based on Measurement Error Models, 2nd de Gruyter.
Pan, V.Y., 2001. Computation of approximate polynomial GCDs and an extension. Information and Computation 167, 71–85. doi:10.1006/inco.2001.3032.
Qiu, W., Hua, Y., Abed-Meraim, K., 1997. A subspace method for the computation of the GCD of polynomials. Automatica 33, 741–743. doi:10.1016/S0005-1098(96)00244-0.
Rosen, J.B., Park, H., Glick, J., 1998. Structured total least norm for nonlinear problems. SIAM J. Matrix Anal. Appl. 20 (1), 14–30. doi:10.1137/S0895479896301662.
Shklyar, S., Kukush, A., Markovsky, I., Van Huffel, S., 2007. On the conic section fitting problem. Journal of Multivariate Analysis 98, 588–624. doi:10.1016/j.jmva.2005.12.003.
Terui, A., 2013. GPGCD: An iterative method for calculating approximate GCD of univariate polynomials. Theoretical Computer Science 479, 127–149. doi:10.1145/1576702.1576750.
Usevich, K., Markovsky, I., 2014. Variable projection for affinely structured low-rank approximation in weighted 2-norms. J. Comput. Appl. Math. 272, 430–448. doi:10.1016/j.cam.2013.04.034.
Van Huffel, S., Park, H., Rosen, J.B., 1996. Formulation and solution of structured total least norm problems for parameter estimation. IEEE Trans, Signal Process, 44 (10), 2464–2474. doi:10.1109/78.539031.
Zeng, Z., 2011. The numerical greatest common divisor of univariate polynomials. Contemp. Math. 556, 187–217. doi:10.1090/conm/556/11014.